Master's Thesis

Automatic Colorization of Webtoons Using Deep Convolutional Neural Networks

딥컨볼루션을 이용한 웹툰 자동 채색

February 2018

Seoul National University
Graduate School
Computer Science and Engineering

Ceyda Çınarel

Automatic Colorization of Webtoons
Using Deep Convolutional Neural Networks

딥컨볼루션을 이용한 웹툰 자동 채색

지도교수 장병탁

이 논문을 공학석사 학위논문으로 제출함

2017 년 11 월

서울대학교 대학원

컴퓨터 공학 전공

Ceyda Çınarel

Ceyda Çınarel의 공학석사 학위논문을 인준함

2017 년 12 월

위 원 장	강유	(인)
부위원장	장병탁	(인)
위 원	김건희	(인)

Abstract

In this work we look at the task of colorizing black and white images on a new domain: webtoons. Although the colorization task has previously been explored for natural images, this type of dataset hasn't been used before. Comics like webtoons present some additional challenges over natural images, such as occlusion by speech bubbles and text. Webtoons are usually produced in color, making them a good dataset for analyzing different colorization models. First we look at some of the previously introduced models' performance on this task and establish a baseline. Then propose a new model to address the problems of color bleeding and color inconsistency. The proposed Color & Apply network is composed of two networks; one network generates sparse color information and a second network uses this generated color information as input to apply color to the whole image. These two networks are trained end-to-end. The proposed model solves some of the problems observed with other architectures, resulting in better colorizations.

Keywords: deep convolutional neural networks, U-Net, encoder-decoder, autocolorization, webtoons

Student Number: 2015-23303

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biologically inspired neural networks such as convolutional neural networks have helped solve many problems in computer vision. Neural networks have even surpassed human level accuracy in tasks such as classification. But when we look beyond tasks that only output a label, and look at tasks such as image generation, there are still many challenges. Image generation or transformation tasks require not only recognizing low level features but also require understanding semantic features in the larger context. Colorization is one such task.

Colorization can be thought of as a conditional image generation problem. It can also be posed as an image transformation problem where the grayscale version of the image is transformed into a colored one. From a mathematical point of view the purpose is to learn a mapping function f such that $f : R^1 \to R^2$ with values in the interval $[-1, 1]$. To learn this difficult mapping understanding the low level features and higher level semantics of the image is necessary.

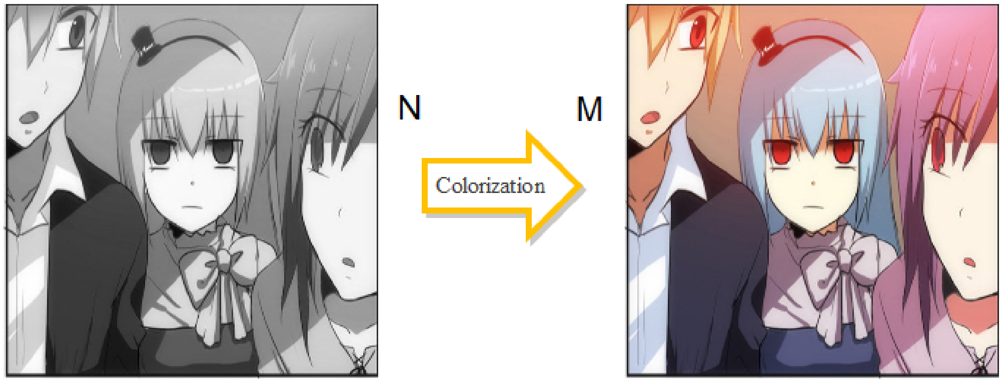


Figure 1.1: Colorization is an N to M mapping problem

1.1 Focus of This Reseach

Purpose of this research is twofold. Training a model capable of generating good possible colorizations while capturing the necessary features for colorization. As well as getting better insight into neural network models used for colorization by evaluating them on a different domain. The task of colorization has so far been only researched on natural images. By looking at this task through the lens of a different domain such as webtoons we hope to gain insight to the inner workings of these models.

1.2 Potential Uses

Any colored image can be thought of as the label of its grayscale version. This opens up the possibility of using colorization task as a way of unsupervised feature learning. [1] shows the colorization task to be better than other methods of unsupervised feature learning such as inpainting.

Colorization is also explored as an avenue for image compression [2]. Although the focus of this work is generating possible colorizations rather than being perfectly faithful to ground truth, this work is an important first step in this direction.

Colorization was also proven useful for downstream tasks such as classification [3, 4]. For example webtoons themselves can have (intentionally) uncolored panels and colorizing these as a preprocessing step can help on tasks such as segmentation. It can be hypothesized that a network trained on webtoons data can learn features useful for other coloring tasks such as coloring sketches implicitly,i.e without needing a reference image.

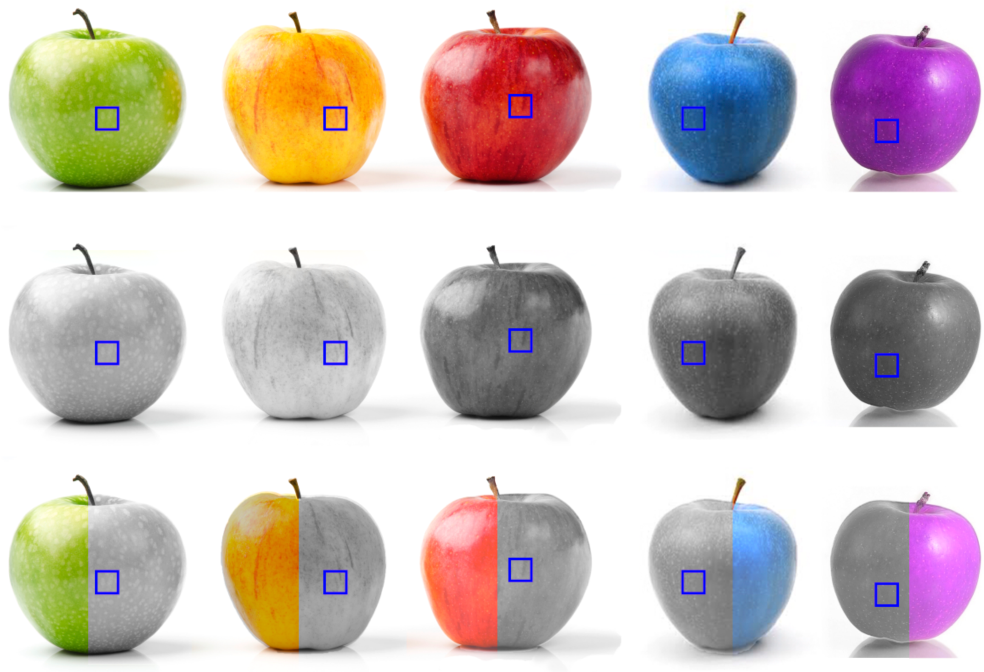


Figure 1.2: Multimodality

1.3 Challenges

Even if the image features are recognized it can be difficult to infer the actual color of an grayscale object, even for humans. For example as seen in Figure 1.2

an apple can be many colors;green,yellow,red,brown... Especially if we think of only a small patch (like the filter size) the distribution of the values may be very similar. Furthermore lightness values can change without effecting the chrominance (seen in bottom row). This multimodal nature of images is a challenge that is inherently present for the colorization task, also making it hard to numerically compare the results of different models.

Compared to natural images colorization of webtoons can be more difficult due to occlusion by speech bubbles, text and texture effects. Some examples can be seen in Figure 1.3. Moreover, comics also show higher multimodality where the possible set of colors for a given entity is potentially only limited by imagination. There can be pink apples in comics but not likely in natural images. There is also more variance in shapes.

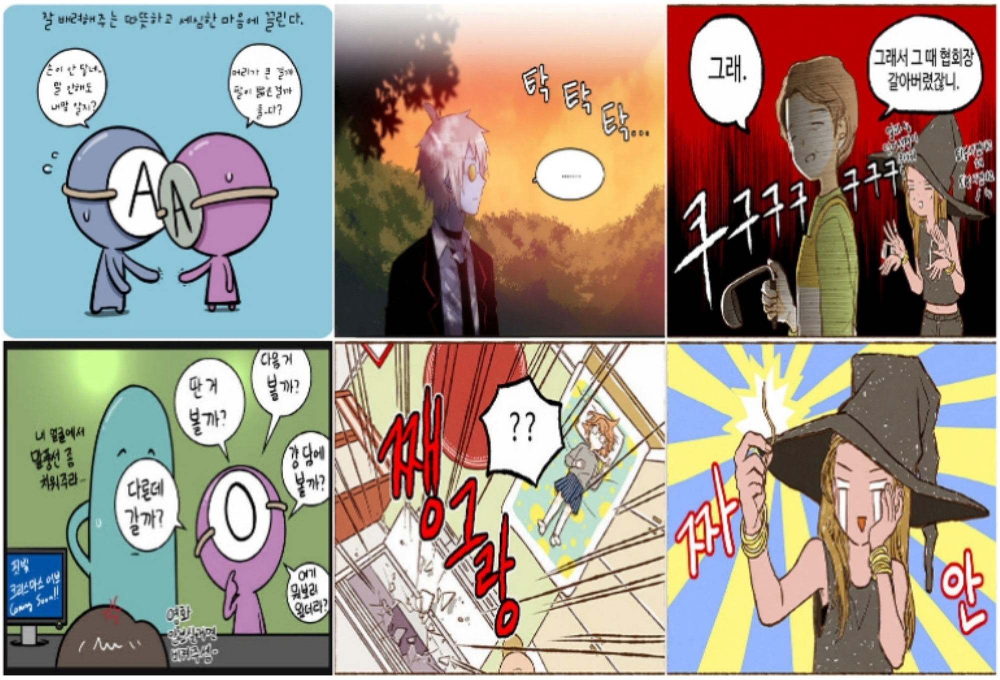


Figure 1.3: Challenging examples from the dataset

There is currently no standard large dataset available for comics as in the case of natural images. For research we have to collect our own data which can be arduous and usually due to copyright sharing is not possible. The lack of pre-trained networks on this type of dataset which can provide important hints to the network such as semantic information also makes comics processing challenging. The work of [5] trained on illustrations for tag classification is the closest to comics domain.

Chapter 2

Related Research

User input guided [6] or fully automatic [7, 8, 9, 3, 4, 10, 11, 12, 13] colorization systems based on neural networks making use of large datasets such as Imagenet [14] has been where the majority of colorization research focused on. To our knowledge there has been no previous works for colorizing comics such as webtoons. Although there are very recent works on colorizing sketches [15, 16, 17, 18, 19, 20, 21] our task is different in that we are colorizing grayscale images and not sketches.

2.1 Colorization of Natural Images

Commonly, pre-trained networks on tasks such as object classification and segmentation are leveraged in the previous works using labeled natural image datasets. This type of auxiliary information is used in [4] when extracting features in form of hypercolumns, [11] uses features from the VGG[22] network, [7] uses additional classification loss,[8] also employ ResNet101 pre-trained on segmentation task. But for our task such pre-trained networks or labeled data is not present.

A key information from [8] is the observation that simply up-scaling the lower resolution version of the chrominance of an image can be sufficient for coloring, meaning only a little color information is needed. They train two networks individually; a coloring network that generates low resolution chrominance channels given grayscale image and a refinement network that combines this chrominance with the grayscale image. [17] also uses two networks trained individually for the task of colorizing sketches. Where a color generating network is trained by removing random patches of color. [9] introduces a model for the more abstract task of image-to-image translation. They use conditional generative adversarial networks(CGAN), conditioning on the input image.

Coloring can also be thought of as a style thus applying style transfer [23, 24] methods for this task also sounds viable. But there are a few things to be considered; style transfer methods such as [23] transfer the whole style, drawing and color. Whereas we would like to leave the drawing style untouched. [25, 26] transfers the drawing style and preserves the color which can be though of as the reverse direction of what we want to do. Color transfer methods [27] based on histogram matching or color mapping between two images exist for tasks like color correction. But these methods rely on there being a reference image given explicitly. [28] does not require specific image pairs but sets of images, success of which depends on the similarity of the two images.

2.2 Colorization of Sketches

When talking about colorization one can also think about the colorization of sketches. Although this is a different type of colorization task where the grayscale (lightness) information is not present. This makes it all the more important to understand the semantic features in an image. That is why most systems rely on coloring based on a reference image [18, 19, 20]. But when the reference image differs in composition from the target image to be colorized, the colorization may fail altogether resulting in grayish outputs. [20] modifies the main architecture introduced in [7] by adding the histogram of a reference image in the fuse layer, followed by post-processing steps. [19] uses the network of [9] on a single reference image, followed by post-processing using segmentation to deal with the extra colorfulness of results generated by conditional GANs [29]. This extra colorfulness is something also observed in our experiments as a result of adversarial loss.

2.3 Lab colorspace

The colorspace to represent the data in is also an important choice. Typically in image processing color images are represented as three dimensional tensors in RGB colorspace. Due to the definition of the RGB colorspace channels of the image are correlated with each other, each channel containing the lightness information. This makes it ill-suited for a task such as colorization. Whereas colorspaces such as YUV, HSV and Lab (also known as CIELAB and L*a*b) allow easily separating the chrominance of an image from its lightness values. Lab colorspace was developed based on human vision and its channels are close to independent [27]. Thus following the literature on colorization, in all the experiments the images are represented in the Lab colorspace. In order to keep the full range of colors we do not quantize (bin) the colorspace unlike [4, 3, 8].

2.4 The Baseline Architecture

Comparing the effects of different loss functions and architectures on the webtoons dataset was necessary to determine a competitive baseline. In [9] a PatchGAN discriminator is used and the loss function is defined as adversarial loss plus L1 loss. While many other works use L2 loss only; with [7, 3, 4] or without [11] some modifications.

Adversarial loss - Although using adversarial loss in addition to the L1 loss might be better for the more general task of image-to-image translation significant benefits weren't observed for the task of colorization. Adversarial loss did generate more colorful results as reported in the original paper [9]. Although this colourfulness may look more satisfying in some cases such as coloring sketches [17, 18] it results in inconsistent coloring. For example while coloring solid color backgrounds as seen in Figure 2.2 top row. On the validation set there were no outstanding perceptual differences and numerically L2 losses throughout the training were also very similar. The average L2 loss on the test set was 1578.8 for the model using only L2 loss, whereas it was 1595.0 for the model with adversarial loss. In both cases, more so with adversarial loss, there was a type of color confusion/uncertainty happening around areas with speech bubbles.

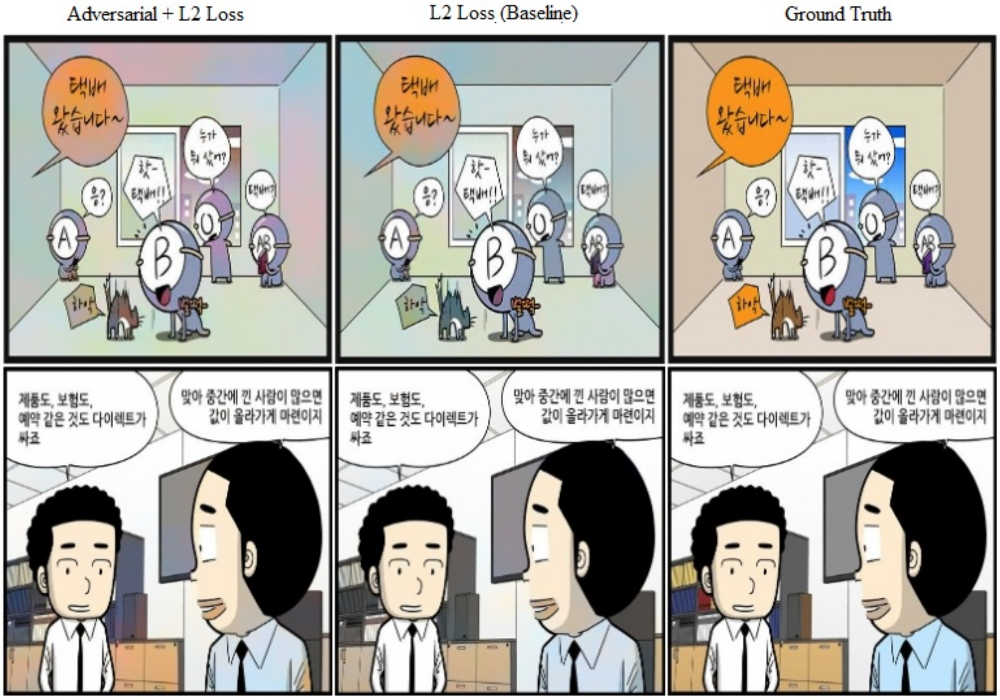


Figure 2.2: Comparison of L2 loss and adversarial loss

From left to right: L2 loss with adversarial loss, L2 loss only, ground truth. Top row; jumbled background colors using adversarial loss. Bottom row: with adversarial loss in some cases it is possible to get less washed out results.

Based on these results the generator architecture used in [9] using L2 loss only is chosen to be a competitive baseline. The details of the layers of network are same as those described in [9] and visualized in Figure 2.1. Each block represents a Convolution-BatchNorm-ReLU layer where first dimension is the feature map count. Filter size is 4 and stride is 2. BatchNorm isn't used in the first layer and ReLUs in the encoder are leaky with slope 0.2. In the variations of this model we will describe in the following chapters, number channels in the first layer is adjusted depending on the input.

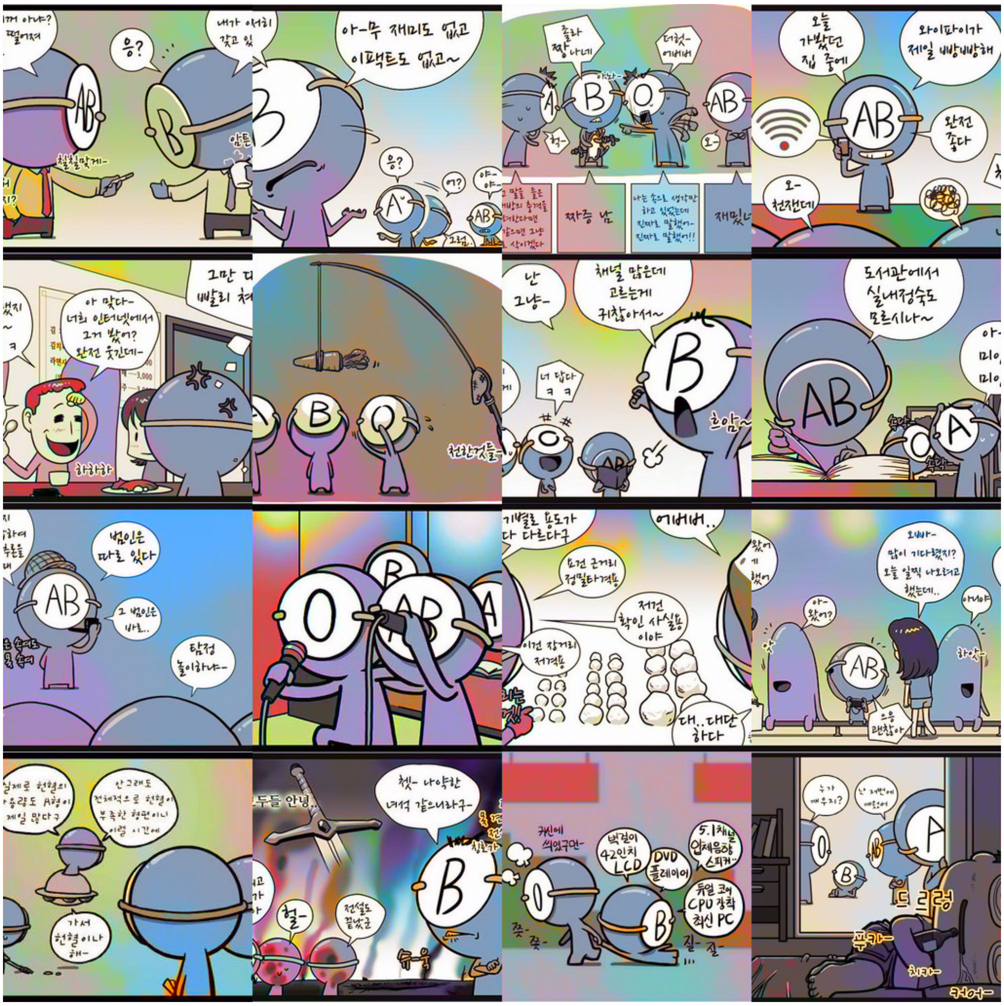


Figure 2.3: Output of the variational auto encoder model

The model of [10] that use variational auto encoders were also trained,results in Figure 2.3. Although trained on a single class for a longer time it didn't reach the results expected. Even though the images were fully colored, there was no consistency and included many highly saturated areas even on the training dataset so we didn't followup on the variations of this architecture.

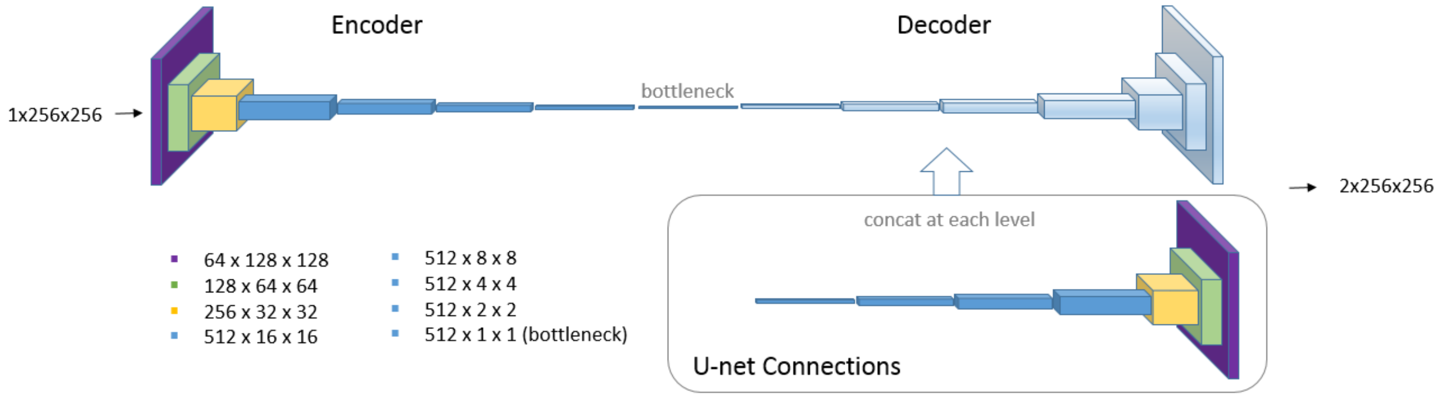


Figure 2.1: The baseline architecture

Chapter 3

Proposed Model: Color & Apply

In this chapter the new proposed model architecture is introduced starting with the process that led to the development of this model.

Different approaches, explained in chapter 5, were experimented with in order to alleviate the problems of jumbled colors and color bleeding we have seen in the results so far. There were many instances where the network would generate the correct color for a small area but fail to apply it to the whole related segment. Another common type of situation was where the correct color hue would be generated but with lower saturation than expected.

To address these problems we first tried an iterative generation process where at generation time the output of the network could be fed back to itself as the input, iterating as many times as needed. To make this network learn how to generate colors, at training time the network was given very sparse (masked) color information. This can be thought of as applying dropout to the inputs but only on the chroma channels. But this single network approach resulted in very saturated colorizations because the output at first iteration contained more color information than the network is trained on resulting on the following iterations being more and more saturated, as can be seen in Figure 3.1.

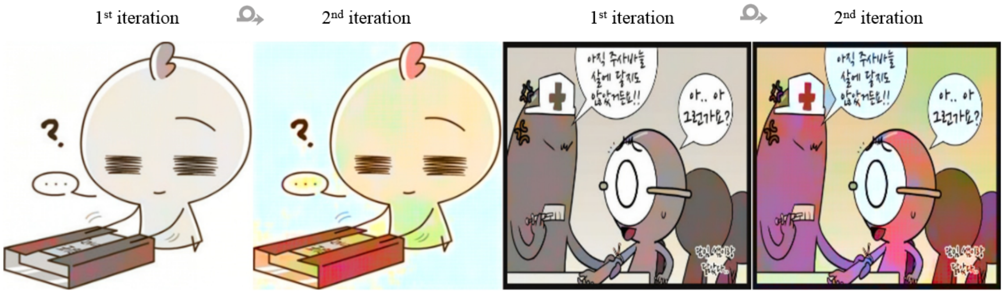


Figure 3.1: Iteratively generated outputs

On the left are images generated at the first iteration, and on the right are images generated at the second iteration by using left image as input. This approach also allows errors made at the first iteration to get propagated. Therefore instead of using one network for both generation and application of color, we propose to train two networks. In this proposed model, one network generates sparse colors and the other network uses this as input to color the whole picture. This can be though of as a person being given a choice of colors for coloring within the lines. We will refer to them as the coloring network and the applier network as it learns how to apply the given colors to the image. An overview of the model architecture can be seen in Figure 3.2. The color generator and the applier networks have the same architecture described in section 2.4 and Figure 2.1

Let's define in_{sparse} to be the input where a random mask is applied to the ab color channels and in_{masked} to be the input where the chroma(ab) channels are fully masked. Masked values are zeroed out thus this can be thought of as applying dropout without scaling. Keep rate of 0.01 is used for in_{sparse}. gen_{sparse} is the image chroma generated by the color generator network using

in_{sparse} and similarly gen_{masked} is generated from in_{masked}. The applier network generates out_{sparse},out_{masked} from in_{sparse},in_{masked} respectively. The total loss function is the L2 loss between three pairs of images. Similar to the baseline loss function the *applier loss* is the L2 loss between the final result with the ground truth $L2(out_{masked}, in_{ground})$. *Color loss* is $L2(out_{masked}, out_{sparse})$. *Color generator loss* is $L2(gen_{sparse}, in_{ground})$.

This method is somewhat similar to that of [17, 9]. Differently from these models the proposed model generates sparse color information rather than a low resolution one. Also differently, the two networks are trained end-to-end. In [9] a pre-trained ResNet-101[30] on segmentation is used, currently this model does not employ, but may also benefit from, such segmentation information. The method of removing random patches for training used in [17] was only able to produce sepia tones when we tried to replicate its results using its open sourced code.

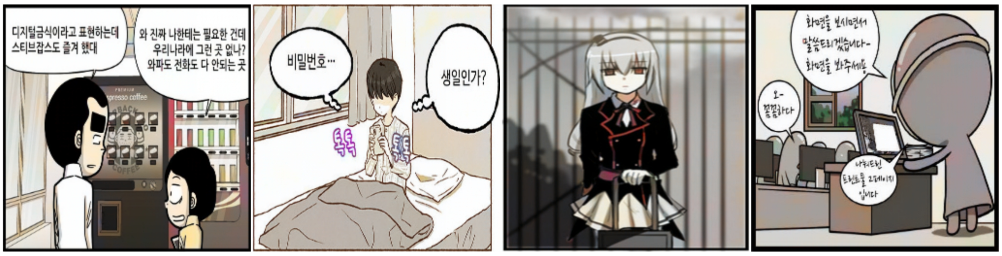


Figure 3.3: Results of training with sparse color information

One interesting result of training the network on sparse color information was that it led the network to focus on the less commonly occurring colors in the smaller details, seen in Figure 3.3. We also tried dropping out on the channel level, omitting one channel at a time, which learned the more general colors but was worse in detailed comparison.

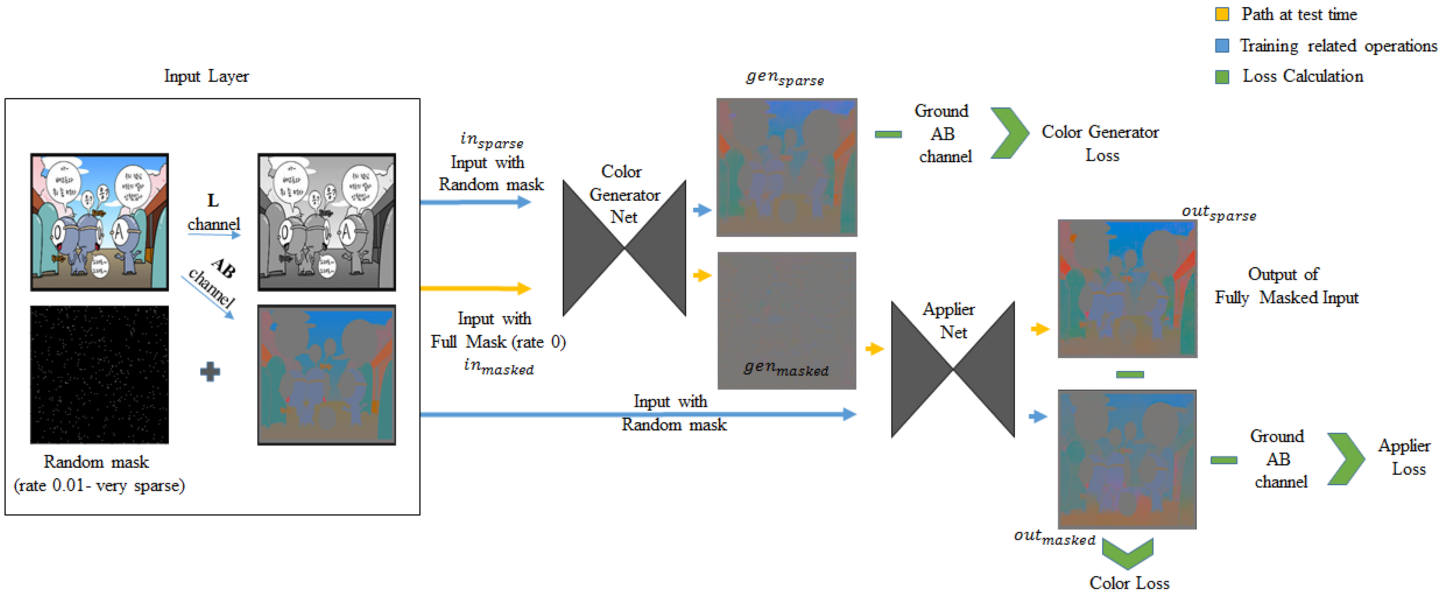


Figure 3.2: The proposed model architecture

Chapter 4

The Dataset

Webtoons are an online form of comics that have gained significant popularity with the spread of mobile devices. Usually released on a weekly schedule and produced in full color they can be the work of a single artist or the work of multiple artists working together; an illustrator, a colorist... They are tailored for smart phones thus stories are told in a vertical layout. A quick scroll can give us a sense of that webtoon's individual visual style. Apart from the drawing style we can easily see the color pallet being used distinctly reflecting the identity of that webtoon. Since webtoons are produced in color this makes them a good testbed dataset for evaluating various colorization models. For the reasons we mentioned above we chose to work on webtoons. Unfortunately, there was no such dataset readily available at the time. Thus we chose to collect our own dataset by scraping Naver webtoons [1].

[1] http://comic.naver.com/index.nhn

4.1 Preprocessing

As with any scraped raw data some preprocessing was done using the following steps:

Firstly, because the scraped images contained multiple panels per image they were cropped into panels using a method based on the common background color. Although this did not always produce exact results it was mostly accurate.

Further to get rid off any meaningless remnants made by the cropping process, images smaller or larger than 200 and 800 pixels in width or height were removed. Since webtoons are usually intended for being read on a mobile device panel sizes are more or less consistent.

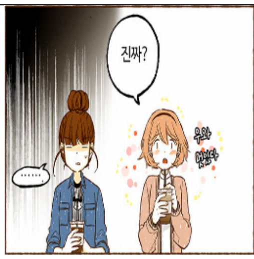
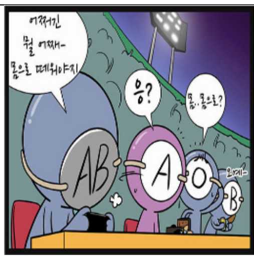
Between each step duplicate images were removed since title and ending panels are repeated. And as a final step, naturally grayscale images were removed which got rid off things like panels with speech balloons only. No additional pruning of the data was done, keeping varied challenging images.

4.2 Properties

An overview of the dataset can be seen in Table 4.1. Since consecutive panels maybe very similar the train test data split was done based on different issues of the webtoon. Otherwise we may overestimate the success of a model since images with very similar composition can end up split between the train and test set.

These five webtoons were picked as our dataset for highlighting some of the potential difficulties of the task mentioned as seen in Figure 1.3; high multimodality, occlusion, texture... The dataset contains; similar/same looking characters with different colors, fine color gradients in the light effects, panels with screen-tones only or with sparse color, complicated or simple solid backgrounds, colored speech balloons... By training using a large webtoon collection, compared to using a single webtoon, the network can generalize better enabling it to color the types of panels that might be uncharacteristic for a specific webtoon alone. We collected a larger dataset that consists of 25 different webtoons but decided to run our experiments first with this smaller dataset to investigate the characteristics of the possible fail cases made by various models in order to find the proper architecture for this task. Common failures of models include; missing small details in effects and accents, failing in presence of texture and occlusion, complicated backgrounds, color bleeding outside the lines. Some examples are shown in Figure 5.4.

Table 4.1: Overview of the dataset

Webtoon Name	Example Images		Data sizes
가우스전자 시즌3 by 곽백수 ©			Train:1756 Val:98 Test:97
슈퍼 시크릿 by 이온©			Train:5691 Val:316 Test:316
일상날개짓 by 나유진 ©			Train:1056 Val:59 Test:58
크리퍼스쿨 by 밀치/얍치 ©			Train:2468 Val:137 Test:137
혈액형에 관한 간단한 고찰 by 박동선 ©			Train:2947 Val:164 Test:163

Chapter 5

Experiments

In section 2.4 we looked at the results of [9] as a baseline. In this chapter we will analyze and compare the results generated by the proposed model. This chapter also covers methods tried prior to the development of the proposed color & apply network. Although some of those models did not generate better results they were important in making key observations that led to the development of the proposed color & apply network architecture where best results were achieved. In line with the purpose of this work results are evaluated on whether the generated coloring is consistent with that webtoons style and free of artifacts described before. All the results in the figures are from the test dataset, with the exception of Figure 2.3 and 3.3. All models were trained for their best results, decided on the validation set for fair comparison. Differences between images are best viewed in fullscreen.

Conditioning on the label - Since knowing which webtoon an image belongs to can reduce the set of possible colors, conditioning also on the label with the adversarial model was tested. But this didn't improve and in some cases decreased the quality. I hypothesize that this is because the model was already able to infer the label without being conditioned on it, and this explicit conditioning decreased the model capacity by a small margin.

Classification loss - Another loss function which is calculated by running the generated image through a VGG16 classifier network was added to the L2 loss. This VGG16 network, explained in more detail below, was trained to classify colored webtoon images. One thing to note here is that this method is highly dependent on the quality of the classifier used. In these experiments no benefits were observed from adding this loss, probably due to the classifier being able to classify the image easily. A classifier more sensitive to the color might be necessary for this approach to work.

Figure 5.1: Activations of a single feature map from different layers of the VGG16 network trained on classification task

5.1 Using Features from Pre-trained Networks

As mentioned in the review of the related research, pre-trained networks are often used for adding semantic information to a model. In our preliminary work we used pre-trained weights of [5] with the model of [11]. Because the work of

[5] classifying tags for illustrations is closer to our domain than natural images. But didn't see significant gains compared to training from scratch.

We trained ResNet-50 and VGG16 models on webtoon classification tasks. There are multiple choices as to how to train this auxiliary network best for it to select features useful for the colorization task. So two versions of each were trained; one on color images another on grayscale. There wasn't a significant difference in classification accuracy between these two versions. We chose to train on grayscale images as this would be the type of image available on inference time. Looking at the activations the ResNet-50 [30] network didn't learn any features beyond the first few layers even on the larger dataset with 25 classes. This is probably due to the task being too simple for training such a deep network. ResNet-50 also achieved lower accuracy compared to VGG16 when the number of classes were 5.

Overall the features learned by VGG16 [22] network architecture were empirically found to be better suited. Activations of a single feature map from different layers of the VGG16 can be seen in Figure 5.1. The outputs of layers *conv_1_2*, *conv_2_2* and *conv_3_1* were used by concatenating them to the encoder at the suitable layers. Looking at the results on the test set there was some minor perceptual improvements on some cases. The average L2 loss was also slightly lower 1561.3 compared to 2.4. Although this model showed some potential improvements it was still not much better than the baseline.

5.2 Results

In this section we will look at the results of the proposed model. The proposed model was especially effective on reducing the multiple color effect, color inconsistency, observed on backgrounds. Comparison between the proposed model and the baseline model using L2 loss only can be seen in Figure 5.2.

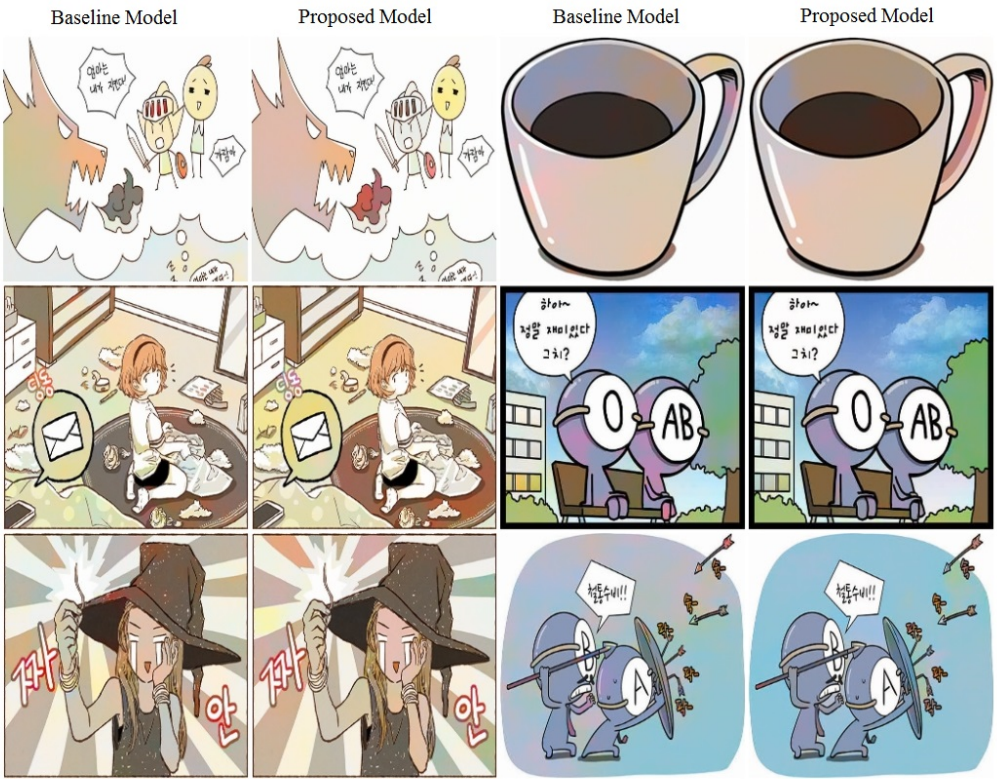


Figure 5.2: Comparison of results; baseline versus proposed model

Images on the left are generated by the baseline model, and images on the right side are generated by the proposed color & apply network. The color & apply model is better at generating more consistent colors for continuous regions with solid color and reduces color bleeding. The amount of dropout applied while training the applier network effects the color information learned by coloring network indirectly through color loss (Figure 3.2).

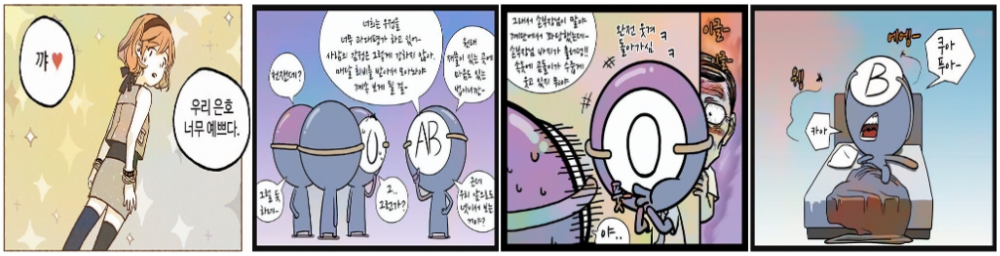


Figure 5.3: Typical mistakes the proposed model makes

Typical mistakes of the color & apply network can be seen in Figure 5.3; segments(1st and 2nd pictures) or boundary like areas(4th and rightmost picture) are colored differently. In the 3rd picture the green segment is colored wrong due to segment shape similarity

Figure 5.4: Common types of mistakes made by all models

1st column; although models successfully color that character's hair in other panels (below) they fail on this instance due to the applied grainy texture. 2nd column; Occlusion results in murky coloring, although successfully colored when fully visible (below). 3rd column; Fails on multimodality, results in the most common colors being selected for coloring. 4th column;text and effects contains very similar light tones only. 5th column; complicated backgrounds (ground truths seen in Figure 1.3)

In general, the U-net [31, 9, 18] structure that adds skip connections between encoder and decoder was very important without these connections models weren't able to generate good colorizations.

5.2.1 Human Evaluation

The numerical evaluation of the results is difficult due to multimodality. Thus human evaluation was conducted. Participants were asked to chose between two images, generated by the baseline and the proposed model, using the web interface prepared seen in Figure 5.5. The images show were from the test set. The users were asked to evaluate based on color consistency, color bleeding and personal preference. The users also had the option to view the ground truth and grayscale version of the image and could choose neither(skip) image if they deemed the results too similar. The image position was randomized for each pair shown to prevent bias. Out of the 20 participants 13 preferred colorizations from the proposed model choosing it %32.2 times more over the baseline while 1 participant showed equal preference.

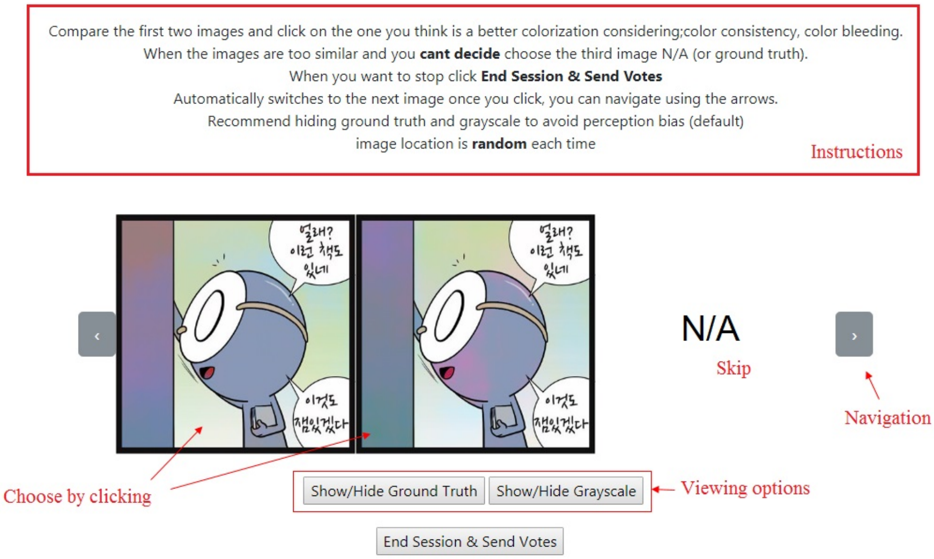


Figure 5.5: The web interface used for evaluation

5.2.2 Outline Colorization

Outline colorization was attempted in order to evaluate the model's capabilities further. Even though the model wasn't trained on outline colorization it was able to generate colors. We can see in Figure 5.6 the proposed model colors within the lines unlike the baseline model. Although there is no grayscale information other than the lines it even manages to produce the correct color in some instances, such as Figure 5.6 top row, meaning that the model learns some semantic information. It should also be noted the outline generation is imperfect and creates a lot of artifacts(noise) especially for complex drawings.

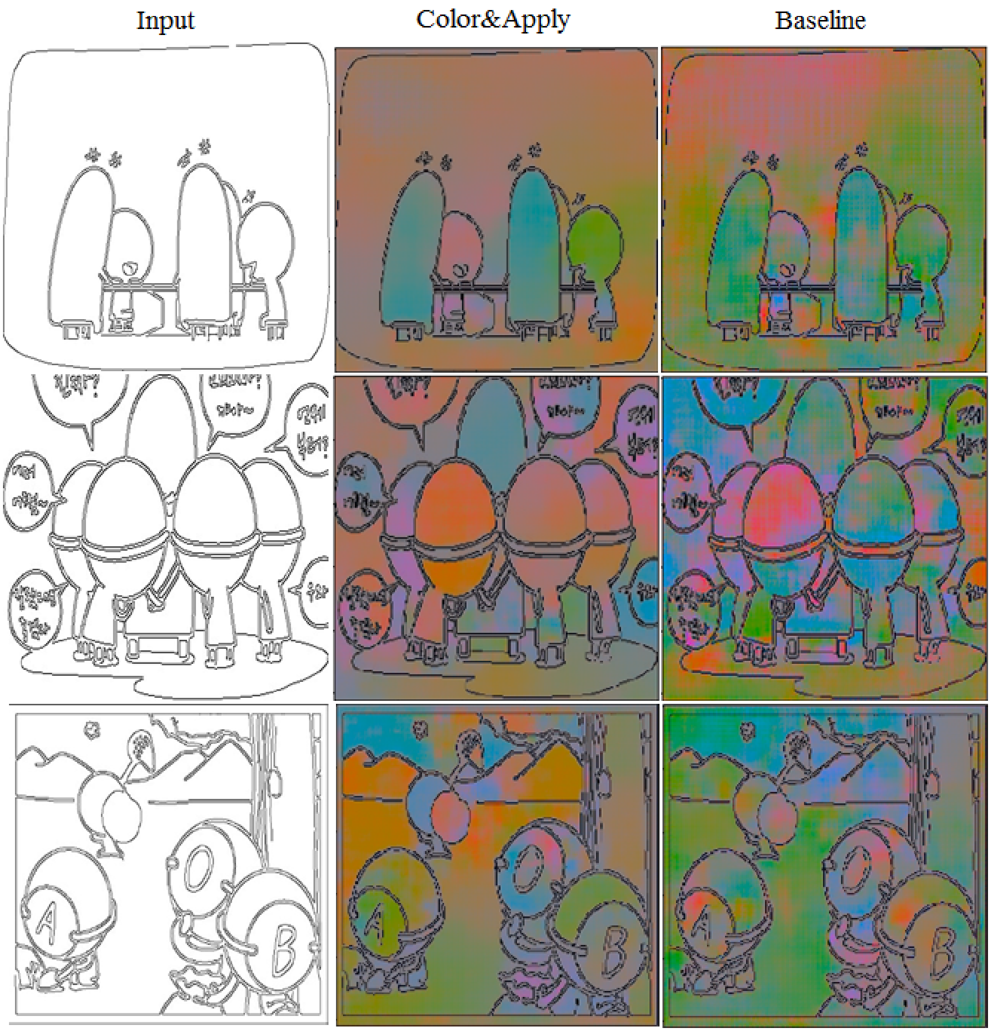


Figure 5.6: Colorization of outlines

Chapter 6

Conclusion and Discussions

This work focused on the task of colorizing webtoons, establishing webtoons as a worthwhile and challenging dataset. We covered the existing models and brought to attention the potential of using a novel dataset from a different domain. Through this approach, we were able to see the artifacts created by different models and gained a deeper understanding of these networks which helped develop a new model architecture. Better colorizations were generated by the proposed model that divide the colorization problem into two parts; the color generation and the color application. Because it is important to also point out the examples the models fail on we shared some instances that fell short of expectations. Training using dropout also for the lightness channel would be a simple extension of this model, potentially allowing the use of this network (features) as a pre-trained network for colorizing sketches. Another interesting extension would be using alternative data sources, other than webtoons, such as frames from an animation where one might even consider leveraging the sequential structure of the data.

Appendix A

Implementation details

All data preprocessing was done with ImageMagick®. In all the experiments images were resized to 256x256. Conversions between different color spaces were done using OpenCV library and values were normalized to be in interval $[-1, 1]$. Mixing different libraries for this purpose should be avoided as numerical ranges used to represent the images may not be consistent between libraries. The precision loss due to conversions between integer and float is another point to be careful of. All experiments were implemented with Tensorflow and run on Nvidia Titan X GPU.

Bibliography

[1] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," *arXiv preprint arXiv:1611.09842*, 2016.

[2] M. H. Baig and L. Torresani, "Multiple hypothesis colorization and its application to image compression," *Computer Vision and Image Understanding*, 2017.

[3] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conference on Computer Vision*. Springer, 2016, pp. 649–666.

[4] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *European Conference on Computer Vision*. Springer, 2016, pp. 577–593.

[5] M. Saito and Y. Matsui, "Illustration2vec: a semantic vector representation of illustrations," in *SIGGRAPH Asia 2015 Technical Briefs*. ACM, 2015, p. 5.

[6] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *arXiv preprint arXiv:1705.02999*, 2017.

[7] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification."

[8] S. Guadarrama, R. Dahl, D. Bieber, M. Norouzi, J. Shlens, and K. Murphy, "Pixcolor: Pixel recursive colorization," *arXiv preprint arXiv:1705.07208*, 2017.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint arXiv:1611.07004*, 2016.

[10] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu, "Unsupervised diverse colorization via generative adversarial networks," *arXiv preprint arXiv:1702.06674*, 2017.

[11] R. Dahl, "Automatic colorization," 2016.

[12] A. Deshpande, J. Lu, M.-C. Yeh, and D. Forsyth, "Learning diverse image colorization," *arXiv preprint arXiv:1612.01958*, 2016.

[13] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 415–423.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 248–255.

[15] Y. Güçlütürk, U. Güçlü, R. van Lier, and M. A. van Gerven, "Convolutional sketch inversion," in *European Conference on Computer Vision*. Springer, 2016, pp. 810–824.

[16] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," *arXiv preprint arXiv:1612.00835*, 2016.

[17] K. Frans, "Outline colorization through tandem adversarial networks," *arXiv preprint arXiv:1704.08834*, 2017.

[18] L. Zhang, Y. Ji, and X. Lin, "Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan," *arXiv preprint arXiv:1706.03319*, 2017.

[19] P. Hensman and K. Aizawa, "cgan-based manga colorization using a single training image," *arXiv preprint arXiv:1706.06918*, 2017.

[20] C. Furusawa, K. Hiroshiba, K. Ogaki, and Y. Odagiri, "Comicolorization: Semi-automatic manga colorization," *arXiv preprint arXiv:1706.06759*, 2017.

[21] Y. Liu, Z. Qin, Z. Luo, and H. Wang, "Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks," *arXiv preprint arXiv:1705.01908*, 2017.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.

[24] ——, “A neural algorithm of artistic style,” *arXiv preprint arXiv:1508.06576*, 2015.

[25] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman, “Preserving color in neural artistic style transfer,” *arXiv preprint arXiv:1606.05897*, 2016.

[26] F. Luan, S. Paris, E. Shechtman, and K. Bala, “Deep photo style transfer,” *arXiv preprint arXiv:1703.07511*, 2017.

[27] H. S. Faridul, T. Pouli, C. Chamaret, J. Stauder, E. Reinhard, D. Kuzovkin, and A. Trémeau, “Colour mapping: A review of recent methods, extensions and applications,” in *Computer Graphics Forum*, vol. 35, no. 1. Wiley Online Library, 2016, pp. 59–88.

[28] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *arXiv preprint arXiv:1703.10593*, 2017.

[29] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2015, pp. 234–241.

초록

본 연구에서는 흑백 웹툰 이미지를 채색하는 방법을 다룬다. 기존의 채색 연구는 주로 자연 이미지를 사용했지만 웹툰 데이터에서는 연구된 적이 없다. 웹툰과 같은 만화 이미지는 말풍선과 텍스트로 이미지가 가려지는 현상 등의 특수한 특성을 가지고 있다. 웹툰은 일반 만화책과 다르게 대부분 채색이 되어 있기 때문에 채색 연구에 활용되기 좋은 데이터이다. 먼저 기존의 채색 모델을 웹툰 데이터에 사용하여 기본 성능을 제시하였다. 색이 경계선을 넘는 현상과 색이 일관되지 않는 문제를 해결하기 위해 새로운 모델을 제안한다. 이 모델은 희박한 색 정보를 생성하는 네트워크와 이렇게 생성된 정보를 입력으로 받아서 전체 이미지의 색을 구하는 네트워크로 구성된다. 이 두 개의 네트워크는 end-to-end로 학습되며 기존 모델에서 발견되는 문제점들을 해결하고 더 좋은 결과를 낸다.

주요어: 딥컨벌루션 네트워크, U 네트워크, 인코더-디코더, 자동 채색, 웹툰

학번: 2015-23303